

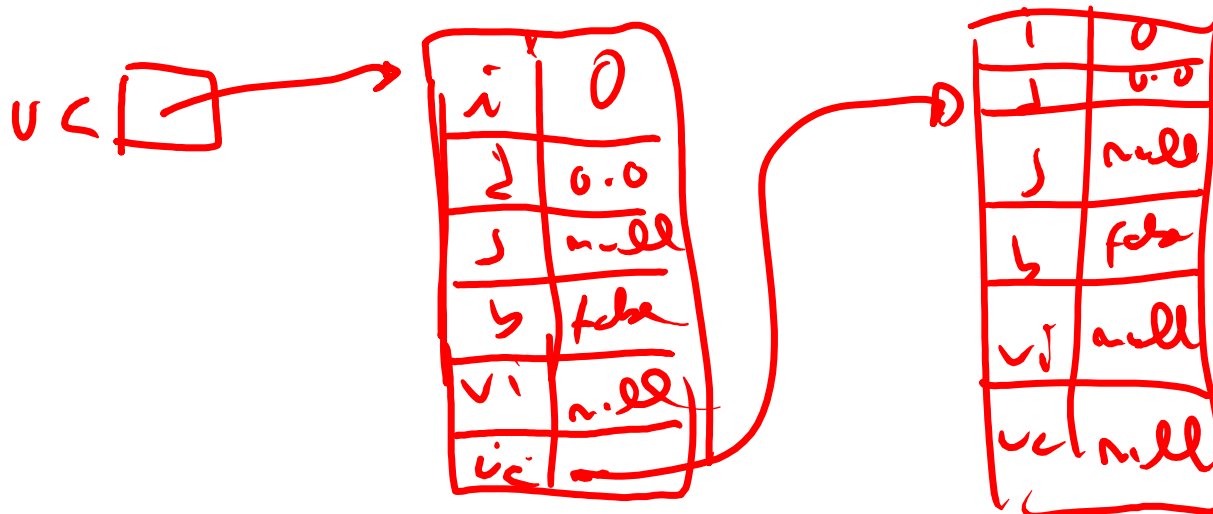
Computação II – Orientação a Objetos

Fabio Mascarenhas - 2016.2

<http://www.dcc.ufrj.br/~fabiom/java>

Classes

- Uma das unidades básicas de um programa Java
- No nível mais simples é como uma *struct* de C, agrupando diversos valores em uma mesma entidade, e possui *campos*
- Campos de uma classe podem ter qualquer tipo, inclusive outras classes
- *Instâncias* de uma classe também são criadas com `new`



Funções e variáveis globais

- Uma *função global* é declarada dentro de uma classe com as palavras-chaves `public static`, seguidas pelo tipo de retorno, o nome da função, sua lista de parâmetros e seu corpo, como em C:

```
public static int mdc(int a, int b) {  
    int q = a / b; int r = a % b;  
    if(r == 0) return b; else return mdc(b, q);  
}
```

- Uma *variável global* é também é declarada com `public static`, seguidas pelo tipo da variável, seu nome, e opcionalmente a inicialização da variável:

```
public static double PI = 3.1415927;
```

- Funções e variáveis locais podem ser usadas dentro da classe onde estão definidas usando apenas o seu nome
- Podemos usar funções e variáveis de outras classes prefixando o seu nome com o nome da classe: `Math.PI`, `String.format("%f", Math.PI)`

Construtor

- Podemos inicializar os campos de uma nova instância de modo parecido do que fazemos em C, mas esse não é o estilo OO apropriado
- A inicialização dos campos de um objeto é tarefa do seu *construtor*
- A declaração de um construtor se parece com a declaração de uma função em C, mas um construtor sempre tem o mesmo nome da classe
- Podemos passar argumentos para a classe quando usamos `new`, e esses argumentos são os parâmetros do construtor
- Dentro do construtor, a variável especial `this` aponta para o objeto recém-criado que deve ser inicializado

Sobrecarga de construtores

- Uma classe pode querer inicializar objetos de várias maneiras
- Para isso, podemos definir vários construtores, contanto que as assinaturas deles sejam diferentes (número e/ou tipos dos parâmetros)
- Essa “sobrecarga” é bastante comum nas bibliotecas Java

(int, double, String, boolean, int, UmaClasse)
(int, double, String, boolean, int(), UmaClasse)
(int, int, int, int, int, int)
()

Identidade de objetos

- Java também tem as operações de comparação `==` (igual a) e `!=` (diferente de)
- Para números e booleanos elas dizem se dois números ou booleanos são iguais ou não
- Mas, para objetos (incluindo strings e vetores), essas operações dizem se os dois objetos que estamos comparando são *a mesma instância* ou não
- Todo objeto possui uma identidade; quando criamos um objeto com `new`, a identidade dele é diferente de todos os outros objetos, mesmo objetos da mesma classe, com o conteúdo dos campos idêntico